

What is claimed is:

1 1. A processor comprising:

2 a plurality of thread partitionable resources that are each partitionable between

3 a plurality of threads;

4 logic to receive a program instruction from a first thread of said plurality of

5 threads, and in response to said program instruction to cause the processor

6 to suspend execution of the first thread and to relinquish portions of said

7 plurality of thread partitionable resources associated with the first thread

8 for use by other ones of said plurality of threads.

1 2. The processor of claim 1 wherein the program instruction is a suspend instruction.

1 3. The processor of claim 1 wherein said logic is to cause the processor to suspend the

2 first thread for a selected amount of time.

1 4. The processor of claim 3 wherein said selected amount of time is a fixed amount of

2 time.

1 5. The processor of claim 3 wherein said processor is to execute instructions from a

2 second thread while said first thread is suspended.

1 6. The processor of claim 3 wherein said selected amount of time is programmable by at

least one technique chosen from a set consisting of:

providing an operand in conjunction with the program instruction;

blowing fuses to set the selected amount;

programming the selected amount in a storage location in advance of decoding the

program instruction;

setting the selected amount in microcode.

7. The processor of claim 1 wherein said plurality of thread partitionable resources

comprises:

an instruction queue;

a register pool.

8. The processor of claim 7 further comprising:

a plurality of shared resources, said plurality of shared resources comprising:

a plurality of execution units;

a cache;

a scheduler;

a plurality of duplicated resources, said plurality of duplicated resources

comprising:

a plurality of processor state variables;

an instruction pointer;

register renaming logic.

1 9. The processor of claim 8 wherein said plurality of thread partitionable resources

2 further comprises:

3 a plurality of re-order buffers;

4 a plurality of store buffer entries.

1 10. The processor of claim 1 wherein said logic is further to cause the processor to

2 resume execution of said first thread in response to an event.

1 11. The processor of claim 3 wherein said logic is further to cause the processor to ignore

2 events until said selected amount of time has elapsed.

1 12. The processor of claim 1 wherein said processor is embodied in digital format on a

2 computer readable medium.

1 13. A method comprising:

2 receiving a first opcode in a first thread of execution;

3 suspending said first thread for a selected amount of time in response to said first

4 opcode;

5 relinquishing a plurality of thread partitionable resources in response to said first

6 opcode.

1 14. The method of claim 13 wherein relinquishing comprises:

2 annealing the plurality of thread partitionable resources to become larger

3 structures usable by fewer threads.

1
1 15. The method of claim 14 wherein relinquishing said plurality of thread partitionable
2 resources comprises:

3 relinquishing a partition of an instruction queue;

4 relinquishing a plurality of registers from a register pool.

1
1 16. The method of claim 15 wherein relinquishing said plurality of thread partitionable
2 resources further comprises:

3 relinquishing a plurality of store buffer entries;

4 relinquishing a plurality of re-order buffer entries.

1
1 17. The method of claim 13 wherein said selected amount of time is programmable by at
2 least one technique chosen from a set consisting of:

3 providing an operand in conjunction with the first opcode;

4 blowing fuses to set the selected amount of time;

5 programming the selected amount of time in a storage location in advance of

6 decoding the program instruction;

7 setting the selected amount of time in microcode.

1
1 18. A system comprising:

2 a memory to store a plurality of program threads, including a first thread and a

3 second thread, said first thread including a first instruction;

4 a processor coupled to said memory, said processor including a plurality of thread
5 partitionable resources and a plurality of shared resources, said processor to
6 execute instructions from said memory, said processor, in response to
7 execution of said first instruction to suspend said first thread and to relinquish
8 portions of said plurality of thread partitionable resources.

1
1 19. The system of claim 18 wherein said processor is to execute said second thread from
2 said memory while said first thread is suspended.

1 20. The system of claim 19 wherein said processor is to suspend execution of said first
2 thread in response to said first instruction for a selected amount of time, said selected
3 amount of time is chosen by at least one technique chosen from a set consisting of:
4 providing an operand in conjunction with the program instruction;
5 blowing fuses to set the selected amount of time;
6 programming the selected amount of time in a storage location in advance of
7 decoding the program instruction;
8 setting the select amount of time in microcode.

1
1 21. The system of claim 18 wherein said plurality of thread partitionable resources
2 comprises:
3 an instruction queue;
4 a register pool.

1 22. The system of claim 21 wherein said processor further comprises:

2 a plurality of shared resources, said plurality of shared resources comprising:

3 a plurality of execution units;

4 a cache;

5 a scheduler;

6 a plurality of duplicated resources, said plurality of duplicated resources

7 comprising:

8 a plurality of processor state variables;

9 an instruction pointer;

10 register renaming logic.

1 23. The system of claim 22 wherein said plurality of thread partitionable resources

2 further comprises:

3 a plurality of re-order buffers;

4 a plurality of store buffer entries;

1 24. An apparatus comprising:

2 means for receiving a first instruction from a first thread;

3 means for suspending said first thread in response to said first instruction;

4 means for relinquishing a plurality of partitions of a plurality of resources;

5 means for re-partitioning said plurality of resources after a selected amount of

6 time.

1 25. The apparatus of claim 24 wherein said first instruction is a macro-instruction from a
2 user-executable program.

1

1 26. The apparatus of claim 25 wherein said plurality of resources comprises a register
2 pool and an instruction queue.

For Office Use Only